Citation

ACM Transactions on Information Systems (TOIS) >archive
Volume 6 , Issue 3 (July 1988) >toc

A rule-based message filtering system

Author
Stephen Pollock

Publisher
ACM Press New York, NY, USA

Pages: 232 - 254 Periodical-Issue-Article
Year of Publication: 1988
ISSN:1046-8188

doi> http://doi.acm.org/10.1145/45945.214327 (Use this link to Bookmark this page)

> full text > abstract > references > citings > index terms > peer to peer

> Discuss > Similar > Review this Article Save to Binder

> BibTex Format

↑ **FULL TEXT:** Access Rules

pdf 1.74 MB

↑ **ABSTRACT**

Much computerized support for knowledge workers has consisted of tools to handle low-level functions such as distribution, storage, and retrieval of information. However, the higher level processes of making decisions and taking actions with respect to this information have not been supported to the same degree. This paper describes the ISCREEN prototype system for screening text messages. ISCREEN includes a high-level interface for users to define rules, a component that screens text messages, and a conflict detection component that examines rules for inconsistencies. An explanation component uses text generation to answer user queries about past or potential system actions based on Grice's conversational maxims.

↑ **REFERENCES**

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 GRICE, H. Logic and conversation. In Syntax and Semantics: Speech Acts. Vol 3, P. Cole and J. Morgan, Eds.,

Academic Press, Orlando, Fla., 1975, pp. 41-58.

2 JOSHI, A., WEBBER, S., AND WEISCHEDEL, R. Living up to expectations: Computing expert responses. In Proceedings of the American Association for Artificial inteUigence-8 (Austin, Tx., Aug. 1984).

3 KUKICH, K. Knowledge-based report generation. Ph.D. dissertation, Carnegie-Mellon University, Pittsburgh, Pa., May 1984.

4 MALONE, T., GRANT, K., TURBAK, F., BROBST, S., AND COHEN, M. Intelligent information sharing systems. MIT Industrial Liaison Program Report, 3-36-87, Massachusetts Institute of Technology, Cambridge, Mass.

5 MCKEOWN, R. Text Generation. Cambridge University Press, Cambridge, England, 1985.

6 MYER, T.H. Future message system design: Lessons from the Hermes experience. In Proceedings o[Distributed Computing COMPCON 80 (Washington, D.C., Sept.). IEEE, New York, 1980, pp. 76-84.

7 Gerald Barber, Supporting organizational problem solving with a work station, ACM Transactions on Information Systems (TOIS), v.1 n.1, p.45-67, Jan. 1983

8 COHEN, P., DAVIS, A., AND DAY, D. Representativeness and uncertainty in classification system. The AI Magazine, (Fall 1985), 136-149.

9 W Bruce Croft , Lawrence S. Lefkowitz, Task support in an office system, ACM Transactions on Information Systems (TOIS), v.2 n.3, p.197-212, July 1984

10 FIKES, R., AND HENDERSON, D. On supporting the use of procedures in office work. In Proceedings of the First Annual American Association for Artificial Intelligence Conference, 1980, pp. 202-207.

11 T. W. Malone , K. R. Grant , F. A. Turbak, The information lens: an intelligent system for information sharing in organizations, ACM SIGCHI Bulletin, v.17 n.4, p.1-8, April 1986

12 MANN, W. Discourse structures for text generation. USC/Information Sciences Institute, Tech. Rep. RR-84-127, Marina del Rey, Calif., Feb. 1984.

13 MERCER, R., AND ROSENBERG, R. Generating corrective answers by computing presuppositions of answers, not of questions. In Proceedings of Conference of Canadian Society for Computational Studies of Intelligence (University of Western Ontario, London, Ontario, May). Canadian Information Processing Society, Toronto, 1984, pp. 16-19.

14 ZLOOF, M. Office-by-example: A business language that unifies data and word processing and electronic mail. IBM Syst. J. 21, 3, (1982}, 272-304.

↑ CITINGS *5*

Douglas B. Terry, A tour through Tapestry, Proceedings of the conference on Organizational computing systems, p.21-30, November 01-04, 1993, Milpitas, California, United States

David Goldberg , David Nichols , Brian M. Oki , Douglas Terry, Using collaborative filtering to weave an information tapestry, Communications of the ACM, v.35 n.12, p.61-70, Dec. 1992

Douglas Terry , David Goldberg , David Nichols , Brian Oki, Continuous queries over append-only databases, ACM SIGMOD Record, v.21 n.2, p.321-330, June 1, 1992

Satoshi Ichimura , Yutaka Matsushita, A pilot card-based hypermedia integrated with a layered architecture-based OODB and an object-forwarding mail system, Proceedings of the 1992 ACM annual conference on Communications, p.431-438, March 03-05, 1992, Kansas City, Missouri, United States

Milam W. Aiken , Olivia R. Liu Sheng , Douglas R. Vogel, Integrating expert systems with group decision support systems, ACM Transactions on Information Systems (TOIS), v.9 n.1, p.75-95, Jan. 1991

↑ INDEX TERMS

Primary Classification:
I. Computing Methodologies

- I.2 ARTIFICIAL INTELLIGENCE
  - I.2.1 Applications and Expert Systems
    - Subjects: Office automation

Additional Classification:

H. Information Systems
- H.1 MODELS AND PRINCIPLES
  - H.1.2 User/Machine Systems
    - Subjects: Human factors
- H.4 INFORMATION SYSTEMS APPLICATIONS
  - H.4.3 Communications Applications
    - Subjects: Electronic mail

I. Computing Methodologies
- I.2 ARTIFICIAL INTELLIGENCE
  - I.2.7 Natural Language Processing
    - Subjects: Language generation

General Terms:
Design, Human Factors

↑ **Peer to Peer - Readers of this Article have also read:**

Editorial pointers
Communications of the ACM 44, 9
Diane Crawford

News track
Communications of the ACM 44, 9
Robert Fox

Forum
Communications of the ACM 44, 9
Diane Crawford

Object-focused interaction in collaborative virtual environments
ACM Transactions on Computer-Human Interaction (TOCHI) 7, 4
Jon Hindmarsh , Mike Fraser , Christian Heath , Steve Benford , Chris Greenhalgh

New Products
Linux Journal 1996, 27es
CORPORATE Linux Journal Staff

A Rule-Based Message Filtering System

STEPHEN POLLOCK
Bell-Northern Research, Toronto

Much computerized support for knowledge workers has consisted of tools to handle low-level functions such as distribution, storage, and retrieval of information. However, the higher level processes of making decisions and taking actions with respect to this information have not been supported to the same degree. This paper describes the ISCREEN prototype system for screening text messages. ISCREEN includes a high-level interface for users to define rules, a component that screens text messages, and a conflict detection component that examines rules for inconsistencies. An explanation component uses text generation to answer user queries about past or potential system actions based on Grice's conversational maxims.

Categories and Subject Descriptors: H.1.2 [**Information Systems**]: User/Machine Systems—*Human factors*; H.4.3 [**Information Systems Applications**]: Communications Applications—*Electronic mail*; I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems—*Office automation*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language generation*

General Terms: Human Factors

Additional Key Words and Phrases: Cooperative tools, explanation systems, intelligent interfaces, text generation

1. INTRODUCTION

Much computerized support for knowledge workers has consisted of tools to handle low-level functions such as distribution, storage, and retrieval of information. However, the higher level processes of making decisions and taking actions with respect to retrieved information have not been supported to the same degree. There are numerous procedures in the office in which information must be tracked and acted on; for example, secretaries screen mail for their managers, stockbrokers scan quotes and act on the information by contacting clients and completing transactions. A number of these functions could be supported by an intelligent filtering system that acts as an assistant to the office worker.

This paper discusses the ISCREEN prototype system for screening text messages. ISCREEN is a rule-based system that was integrated with an existing office information system. It includes the following components:

—a friendly interface that allows unskilled users to specify instructions to the system in terms of rules;

Author's address: Bell-Northern Research, 522 University Ave., Toronto, Ontario, Canada, M5G 1W7.

—an explanation component that utilizes text generation to produce answers to user queries in English sentences;
—a conflict detection component that evaluates users' instructions for completeness and consistency;
—a screening component that intercepts messages as they are received, decides on actions based on users' rules, and takes the actions it has recommended.

Users of the ISCREEN system are able to describe the types of text messages they receive and provide instructions on what is to be done with them. For example, a user may specify that messages from any managers are important and are to be filed in an *important* box and printed.

The remainder of this paper is divided into four sections. The second section describes related work that has been performed in the area of information filtering in messaging systems. The third section describes the functionality of ISCREEN, in terms of the way that rules are defined, the manner in which ISCREEN deals with incomplete or conflicting instructions, the way that messages are screened, and the types of explanations provided by the system. The fourth section provides technical details on the implementation of ISCREEN. Conclusions are presented in the fifth section.

2. RELATED WORK

One of the key steps in designing a system to act as an intelligent assistant is to decide how much of the task should be taken on by the system and how much should be left to the user. The approach taken in developing the ISCREEN system differs from other approaches taken to the same problem in that ISCREEN attempts to handle more of the task.

The reason for adopting this approach was that the intended users for the ISCREEN system include senior managers who receive large quantities of electronic mail and currently have secretaries to screen it for them. These managers are not regular users of computer systems, and it was felt that the interface to the screening system would have to be very easy to use. One of the goals in developing ISCREEN was to simulate as closely as possible the interaction between the manager and secretary in terms of the manner in which instructions are given and the type of feedback that can be provided in return. To do this effectively, it was found that ISCREEN had to take more responsibility in the screening process.

The first messaging system employing filtering capabilities was Hermes [13]. This system featured the use of specialized templates for creating messages so that a message could be categorized according to the template that was used to create it. Manual filtering of messages was supported by allowing users to specify conditions on message attributes when performing operations such as printing, reading, or filing. For example, a user could file all messages sent after a given date by a given individual in a specified box. Although these activities were performed manually by the user, the developers of Hermes foresaw the possibility of messaging systems handling operations such as these automatically.

The message screening problem was also previously addressed in the Information Lens system [9]. A fundamental difference between Lens and ISCREEN is that ISCREEN attempts to handle more of the task of rule definition than

Lens. Where Lens opens up the knowledge base to the user to be manipulated directly, ISCREEN performs some of this operation itself and hides certain details from the user. The systems work differently in the way that rule conflicts are resolved. A rule conflict is a situation in which two rules act on the same message, when the user's intention is that only one of them should apply.

In the Lens system there are two mechanisms for dealing with this problem. The Lens system allows users to define a series of message templates in a hierarchically ordered frame structure. The templates can be used to generate different message types, and rules can be defined to act on these types. Use of a well-designed frame structure would allow the specification of a set of rules in which few conflicts occur. Conflicts that do occur are handled by allowing the user to specify an ordering of the rules that apply to a given message type. The Lens system itself applies an ordering of rules which act on messages at different levels of the frame structure. In situations where rules do not apply to a specific message type, no mechanism is available for effectively handling conflicts.

The ISCREEN system does not use a frame structure for classifying message types and does not require the user to assign an ordering to rules. Most messages received by ISCREEN users are sent by nonusers, so the message templates would not be used by senders. It was also felt that hierarchical frame structures for the definition of rules and ordering of rules would be too complex for the user group. Instead ISCREEN accepts a simple flat-rule structure, looks for situations in which conflicts could occur, and prompts the user with three choices for resolution of each potential conflict found. In this way, ISCREEN builds the rule hierarchy itself.

Two unique features that allow ISCREEN to do this are the conflict detection component and the explanation component. The conflict detection component compares a new rule to the existing rule base to test for potential conflicts. Although the Lens system features a simple explanation capability to reference rules which acted on a message, it was found that a more elaborate facility was required in ISCREEN given the broader role that it plays in rule definition.

An additional difference between the systems is that ISCREEN may include an added knowledge base which allows users to reference organizational variables in their rules, identifying positions of and relationships between members of the organization. Value matching variables are also used to provide flexibility in rule definition, allowing the user to match a value from one part of the message and use it elsewhere in a rule definition. The use of organizational variables in rules supports the ability to perform social filtering as outlined in [9] where a user can specify actions to be performed on a message based on the position of or relationship to the message receivers or senders.

3. ISCREEN FUNCTIONALITY

3.1 Overview

The ISCREEN system is integrated with an office information system (OIS) that was developed at Bell-Northern Research for use in field trials. The OIS features a messaging system allowing users to send text messages and a filing system allowing them to file messages and other documents in 'boxes' that they

create. Each text message on the OIS consists of an envelope and the message content. The envelope contains information on who the message is from, to, and copied to; what the message is on the subject of; and the date of the message. In addition to names, the department numbers, companies, and locations of individuals are listed in the envelope.

ISCREEN acts as an assistant to users of the OIS by screening text messages received by users. The mail screening process is split into the following four broad activities: accepting instructions, screening mail, taking actions, and providing feedback.

—Users provide instructions to ISCREEN in the form of rules, which include a list of conditions and actions. The conditions describe values associated with attributes of messages (e.g., who the message is from, what it is about). Actions describe what is to be done with messages that match the specified conditions (e.g., forward, file, delete). A special purpose editor is used to define these rules. A unique feature of ISCREEN, the conflict detection component, examines new rules to ensure that they do not introduce conflicts with existing rules.

—ISCREEN intercepts text messages as they are received and makes decisions regarding what should be done with them based on rules provided by each of the recipients. To do this it matches conditions specified in rules against the content and envelopes of messages. It has the ability to resolve variable references in rules. For example, if users specify that messages from their managers are to be filed in an *important* box, the system is able to determine whether a message is from the users' managers.

—ISCREEN has the ability to take actions as recommended by rules. The system makes the appropriate calls to the messaging and filing systems on the OIS to carry out actions. Accordingly, the range of actions handled (e.g., filing, deleting, mailing to distribution lists) is determined by the messaging system that ISCREEN is connected to rather than by ISCREEN itself.

—ISCREEN uses a text generation component to provide various types of explanations to the user. This component is called by the conflict detection component to describe how conflicts can occur between rules and is also called by the query component, which answers questions posed by the user. The user can ask why actions were performed by the system, what the system would do under given circumstances, and can also ask for descriptions of rules.

3.2 Providing Instructions

3.2.1 *Rule Format.* Figure 1 illustrates two examples of rule definitions using the ISCREEN rule editor. The user enters *conditions* in the form on the left side of the screen and *actions* in the form on the right side. The conditions are compared to messages received by the user and can be satisfied by a message that has all the attributes described. The actions specified by the user relate to actions that should be performed on the message or on other specified messages when the conditions are satisfied. *Multiple conditions*, separated by commas, are interpreted as conjunctions (all specified conditions in a rule must be satisfied for the action to be taken).

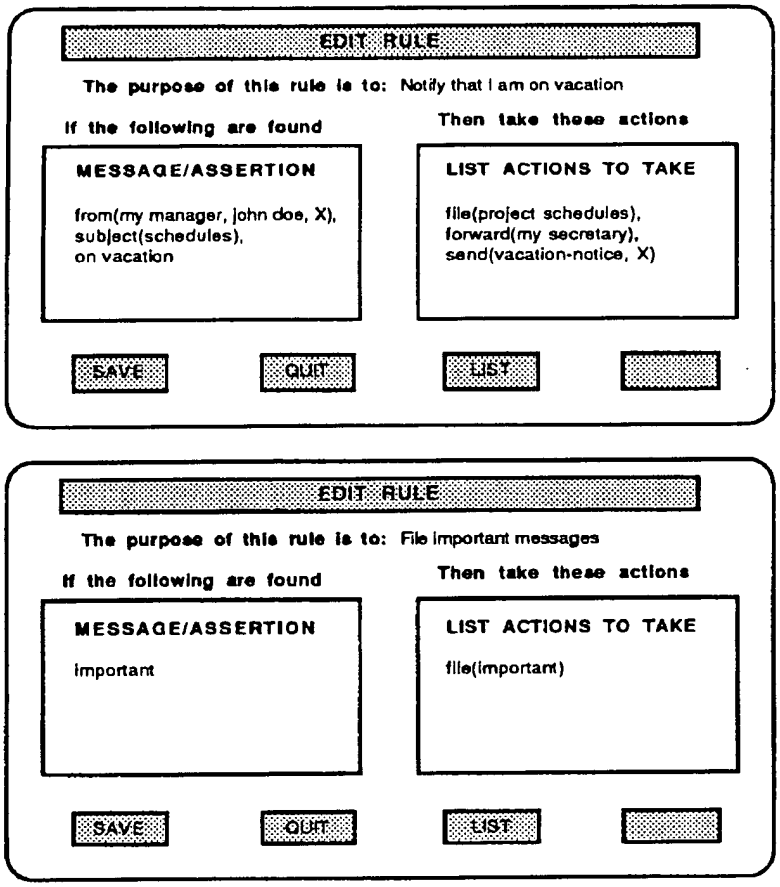

Fig. 1. Rule editor.

The first rule states that if the user receives a message from any managers or John Doe, and if the message contains the keywords "schedules" in the subject, and if the user is on vacation, then the message should be filed in the "project schedules" mailbox, forwarded to the user's secretary, and a message notifying that the user is on vacation should be sent to the sender of the message. In the second example rule, the user states that if an important message is received, it should be filed in the "important" box.

3.2.2 *Condition Definitions.* Three types of condition definitions are supported by the ISCREEN system: *attribute/value pairs*, *message classifications*, and *assertions*. An *attribute/value pair* specifies the name of one of a set of message attributes recognized by the system, plus one or more values that may be associated with the attribute. Recognized message attributes include from (who the message is from), to (who the message is addressed to), copy (who is copied on the message), subject (keywords in the subject line), content (keywords in the content of the message), sent (who the message was sent by), length (length in lines of the message), recipients (the number of recipients of the message), and type (bulletin or regular message). Multiple values associated with a single attribute are separated by commas and are interpreted as disjunctions, so that

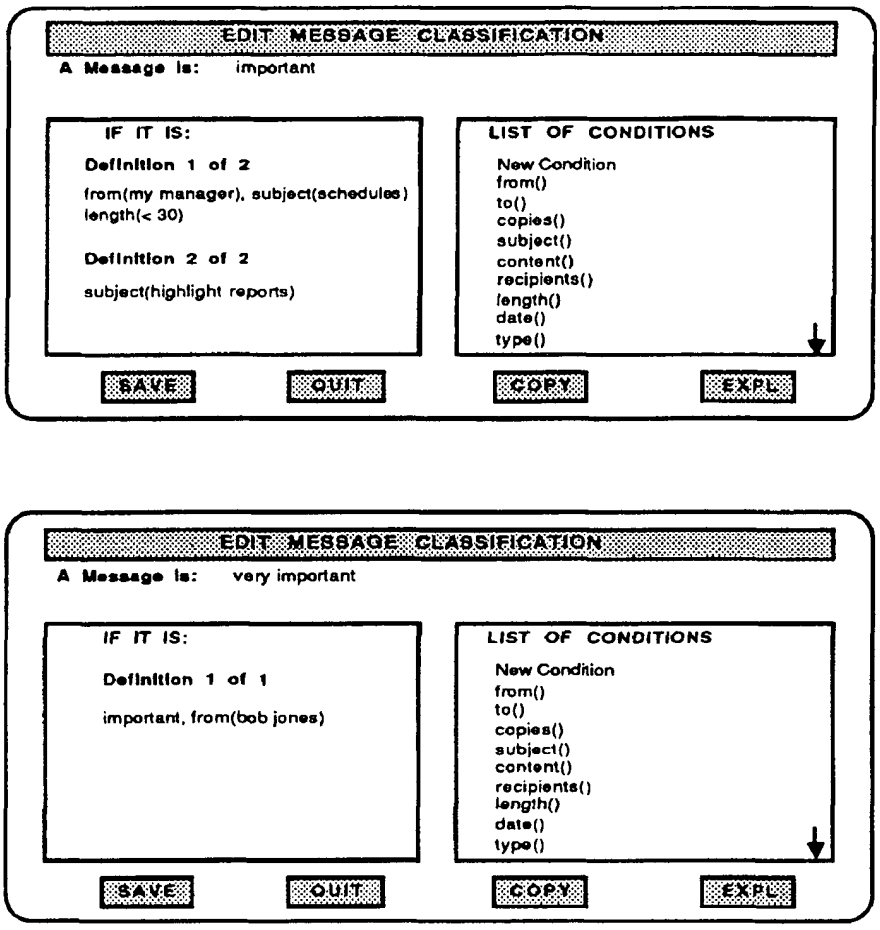

Fig. 2. Message classification editor.

the condition would be satisfied if a message were found whose named attribute matched any one of the specified values.

A *message classification* is a means of collecting other conditions to be reused in different rules. In the second example the rule references a message that is "important." Importance is not an attribute associated with a text message but rather a subjective judgement on the part of the user. For example, a user may consider a message to be important if it is from the user's manager on the subject of schedules or if it is on the subject of highlight reports. Thus a classification of "important" can be defined by specifying a set of *attribute/value pairs* associated with an object. *Message classification* definitions are illustrated in Figure 2.

The advantage of providing an ability to create classifications such as these is that the concept of "important" can be reused in other rules without having to repeat the definition. When the criteria for an important message change, only the classification definition needs to be modified. When this is done any rules referencing the classification will inherit the changed properties.

The third type of condition, the *assertion*, does not describe objects but rather describes other conditions that may have an impact on the manner in which

messages are to be screened. In the first example rule in Figure 1, an assertion "on vacation" is used to identify situations in which the user is on vacation. Users define the assertion with an editor provided by ISCREEN, thus notifying the system that they may go on vacation at some time. Users then have the ability to toggle the assertion on and off in order to notify the system that they have gone on vacation or have returned. *Assertions* provide users with the ability to instruct the filter about what to do under special circumstances.

There are circumstances in which information referenced by rules will only be known when messages are received. To handle these situations ISCREEN supports the use of two types of variables in defining rules: *value matching* and *organizational information.*

A *value matching* variable is used to store the value associated with some attribute of a message. An example is the variable "*X*" specified in the first example rule that is instantiated to the name of the sender of the message. In this example the "from" condition will be satisfied by a message that is from a person who is one of the user's managers or from John Doe. The name of the person who satisfies this condition is then stored in the variable, which is used in the *action* definition to identify the recipient of a notice to be sent. When a *value matching* variable is used without other conditions (e.g., from(*X*)), it matches anything and stores the value. The ISCREEN rule editor recognizes single upper case letters as value matching variables.

Organization information variables are used to reference information about the organization in which the user works. The use of these variables makes it unnecessary to make changes to the rule base because of organizational changes. In the first example rule the filter will attempt to find a message from "my manager." To match this condition the filter will consult a corporate knowledge base in order to determine who the user's manager is. Because a variable is used instead of the manager's name, there will be no need to modify the rule when the user has a new manager. Types of *organization information* variables used in ISCREEN are

—Absolute job positions (e.g., a manager, a secretary)
—Relative job positions (e.g., my manager, my secretary)
—Values representing organizational policy such as due dates.

The use of organizational information variables is supported in ISCREEN through the use of a corporate knowledge base (see Section 4.2).

3.2.3 *Action Definitions.* Actions supported by ISCREEN include forwarding, filing, replying to or deleting a message, or sending another message altogether. The action definition specifies

—the type of action to be performed
—the object(s) that the action is to be performed on
—individuals the action is to be directed toward
—options describing how the action is to be performed.

The use of variables is supported in action definitions, allowing the user to specify individuals to whom the action is to be directed (e.g., forward to my secretary).

3.2.4 *Incomplete Instructions.* One of the initial steps in developing ISCREEN was to interview senior managers to get instructions on how their mail was to be screened. Through this process it was found that the instructions provided were often incomplete. One source of difficulty was that a certain amount of common sense was assumed on the part of the instruction taker. For example, a manager might specify that any messages about project XYZ be forwarded to John Doe. The manager would not also specify that the message should not be forwarded if it is from John Doe. A human assistant would have enough common sense to be able to handle a situation like this without instructions.

Another common difficulty in the instructions provided by users was that incomplete instructions tended to conflict with each other. Consider the following three instructions:

1. If I receive a message from one of my Directors on the subject of highlights then it should be filed in my "highlights" box.
2. If I receive a message that is from Dept 9W20 with the words AI, PROLOG, or screening in the content, then file it in my "AI" box.
3. Messages from my Directors should go into my "directors" box.

All three of these criteria would be satisfied by a message from one of the user's directors in Dept 9w20 which contains a highlight report referencing AI applications. However, it was not the user's intent that the same message be filed in all three boxes. A human assistant can share the responsibility for ensuring that instructions are properly carried out by detecting situations in which they conflict and asking for clarification. An important requirement for the ISCREEN system was that users be able to provide the types of instructions they would provide to human assistants. Because of this requirement, it was necessary to incorporate two types of analyses in the system: *conflict detection* and *sensibility*.

Conflict detection involves examining instructions at the time they are given by the user in order to determine whether there is some inherent conflict within the instruction itself or with the user's other instructions. When ISCREEN detects a conflict, it either makes an assumption regarding the user's intent, or in cases where it cannot determine what the intent was, it asks the user for further instructions. The text generation component is used to generate prompts explaining the nature of the conflicts. The responses produced by ISCREEN to the entry of three rules are listed in Figure 3.

When the first rule is entered (anything from Toronto) no conflicts are detected. When the second rule is entered specifying highlight reports, ISCREEN recognizes that a highlight report could also be from Toronto because the user's director works there. In this case, the conditions matching a highlight report are more specific than those matching something from Toronto, so ISCREEN assumes that the user would want the message filed in the "highlights" box instead of the "toronto" box.

RULE DEFINITION:

1. FILE TORONTO MESSAGES

IF: from(toronto) THEN: file (toronto)

SYSTEM RESPONSE:

The new rule does not create any conflicts.

---

2. FILE HIGHLIGHT REPORTS

IF: from(my director), subject(highlights) THEN file (highlights)

SYSTEM RESPONSE:

The new rule 'FILE HIGHLIGHT REPORTS' and the old rule 'FILE TORONTO MESSAGES' would conflict if a message were from location toronto(because your director works at location toronto), on the subject of highlights. Since 'FILE HIGHLIGHT REPORTS' specifies messages that are a subset of the other rule, I am assuming that it takes precedence.

The new rule creates 1 conflict that I can resolve.

---

3. FILE AI MESSAGES

IF: from(9W20), content(natural language, screener, prolog) THEN: file (ai)

SYSTEM RESPONSE:

The new rule 'FILE AI MESSAGES' and the old rule 'FILE TORONTO MESSAGES' would conflict if a message were from location toronto (because 9W20 works at location toronto), with the content words natural language, screener or prolog. Since 'FILE AI MESSAGES' specifies messages which are a subset of the other rule, I am assuming that it takes precedence.

The new rule creates 2 conflicts, 1 of which I can not resolve.

The new rule 'FILE AI MESSAGES' and the old rule 'FILE HIGHLIGHTS' would conflict if a message were from your director (because your director is in department 9W20), on the subject of project schedule, and with the content words natural language, screener or prolog. Enter resolution using softkeys (NEW, OLD, BOTH).

Fig. 3. Conflict detection.

When the third rule is added, ISCREEN recognizes that a highlight report could also be about AI and asks the user which rule should be followed in this case, the AI rule, the highlights rule, or both. In the third case ISCREEN also recognizes the conflict with the "File Toronto Messages" rule but assumes that the user intends the messages to be filed in the "AI" box because the conditions are more specific.

ISCREEN maintains information about the resolution of each conflict found between pairs of rules. This information indicates the identity of the two rules and which rule(s) should be followed in acting on the message. When rules are deleted, the associated conflict information is removed.

The *sensibility* check is intended to apply judgement in dealing with conditions that may arise at the time a message is being screened. For example, a user may define a rule stating that a message about AI is to be forwarded to the rest of the

user's group. When such a message is being screened, ISCREEN examines the "copy" and "to" list to determine whether some members of the group have already received the message. The check is intended to recognize and veto actions that do not make sense. A set of rules is used in this stage which are similar to the message screening rules.

3.2.5 *Entering Rules.* ISCREEN provides editors for entry of rule and message classification definitions (shown in Figures 1 and 2). Using these editors, conditions and actions can be typed directly into the forms on the screen or can be selected from lists which are displayed by pressing the "LIST" softkey. The "List of Conditions" form shown in the message classification editor is an example of such a list and includes primitive conditions (e.g., from( )), names of user-defined message classifications (e.g., "important message"), assertions (e.g., "on vacation"), and organizational variables (e.g., "my manager").

When a definition is saved, the editor recognizes errors such as unknown conditions or actions and prompts the user with a descriptive message. When the definition has been saved without errors, the conflict detection component is invoked, which may further prompt the user with information about rule conflicts. In each case the nature of the conflict between two rules is described, and the user is prompted with a choice of whether the old or new rule should take precedence, or whether both should be applied.

3.3 Screening Messages

When users access the OIS and enter the messaging service, they are presented with a list of new messages. Pressing an "Action" softkey produces a sorted list indicating actions that have been taken with each message. The user has the option of selectively listing and viewing messages from each category (e.g., all those that have been filed in the "important" box). The user can select from a list of actions performed by the system and request an explanation about why the action was performed as illustrated in Figure 4.

3.4 Explanation

The explanation feature of ISCREEN answers *what if* and *why* questions, describes definitions of rules and message classifications, and explains conflicts that are found in user's rules. A text generation component is used to generate grammatically correct paragraphs in response to user queries. This approach was chosen because it provides a lot of flexibility in the way that responses are structured.

What if questions allow users to test their rules by asking what ISCREEN would do if a message with specific attributes were received. The option to ask the question is provided in ISCREEN's main menu. When asking a *what if* question, the user is presented with the same forms that are used for defining message classifications (shown in Figure 2). Using these forms, the user fills in the description of a message (e.g., important, from(my group)) and presses a "WHAT IF" softkey to ask what would be done if such a message were received. In this way, the same interface is used to describe messages, whether for the purpose of rule definition or explanation.

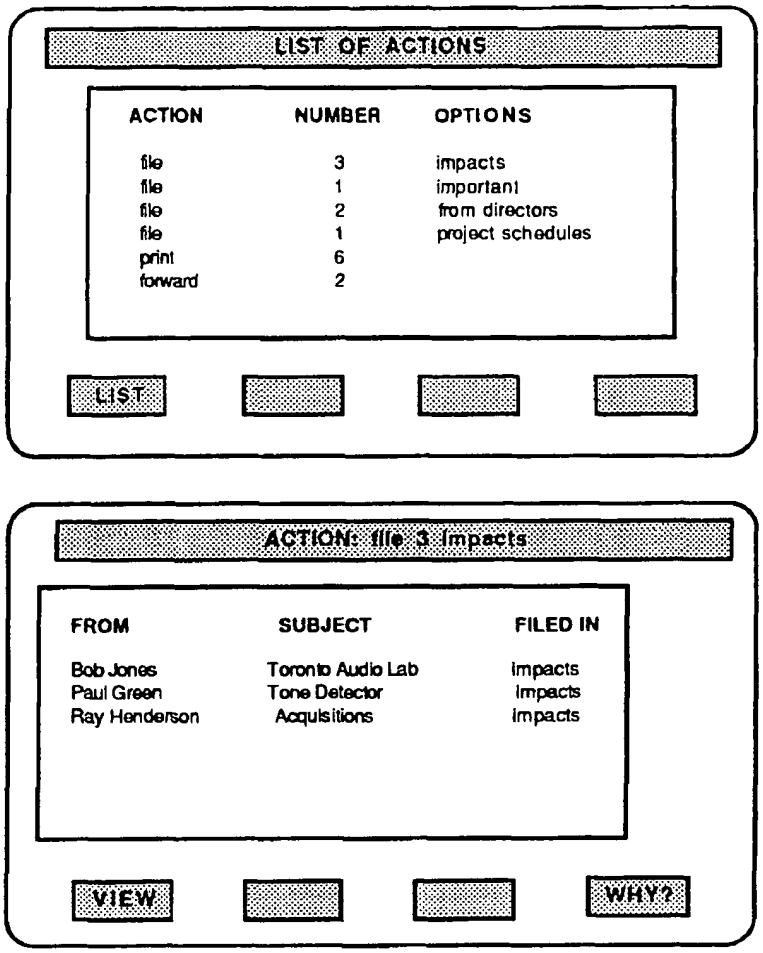

Fig. 4. Action list.

When viewing lists of actions (shown in Figure 4), the user can select a list item and press the "WHY" softkey to ask why an action was performed. Similarly, when viewing a list of rule or message classification definitions, an "EXPLAIN" softkey can be pressed to produce a description of a definition and information on where it is used. When using the rule editor (shown in Figure 1) to save a new definition, the explanation feature is invoked automatically to describe any conflicts between the new rule and existing rules.

The initial version of the explanation system was implemented to produce correct answers to user queries. It was soon found that it was necessary but not sufficient to produce correct answers because a number of the answers were misleading or withheld important information. Enhancements were then made to the explanation feature with the goal of producing answers that did not violate the Cooperativity Principle [5] as expanded in [6]. This principle consists of a set of maxims which guide communications among cooperating individuals. The following examples illustrate some of these enhancements and the reasons for adopting them. The examples are based on the hypothetical *what if* questions and are ordered according to the maxims of the Cooperativity Principle.

Quantity: *The explanation should be as informative as required for the purpose of the information exchange.*

The following example illustrates a reply which does not provide sufficient information but which is nevertheless correct. A user instructs the screener that any messages from John Doe on the subject of project schedules should be filed in an "important" box. If the user asks what the system would do if a message from John Doe were received, the system could correctly answer that it would not do anything.

Although this answer is correct, it does not supply a sufficient quantity of information based on the user's intent in asking the question. A more reasonable response would probably be

> "I would not do anything in this situation. However if the message were also on the subject of project schedules I would file it in your important box."

In formulating responses to queries, ISCREEN first collects a list of items that could be used in the response and classifies them. Two of these items include rules that would be matched by the conditions specified in the user's query and also rules that would be almost matched. ISCREEN seeks to provide a reasonable quantity of information from that list based on assumptions about the user's knowledge level and intent in asking the question. Three cases are recognized in deciding on an appropriate quantity to include in the response:

—If the conditions in the user's query completely match a large number of rules, then the user is told about those rules. In this case ISCREEN assumes that in making this query the user is well informed and will not seek to provide the user with information that is of less relevance.

—If the conditions in the user's query completely match a smaller number of rules, then the user is told about those rules, and a brief reference is made to rules that are almost matched. In this case ISCREEN assumes that the user is fairly well informed but might have a small amount of interest in some of the less relevant information.

—If the conditions in the user's query do not match any of the rules, then a greater level of detail is provided on some of the less relevant information. In these situations ISCREEN assumes that the user is not well informed with the content of all the rules and is simply looking for some information on the rule base.

For example, consider a rule base containing the following instruction:

> If a message is from John Doe about Information Screening, and I am on vacation, then forward it to Bob Jones.

If the user poses a query: "What if I am on vacation and I get a message from John Doe about Information Screening?" ISCREEN provides only the most relevant information because it assumes that the user is well informed. The assumption is based on the fact that the user referenced the specific conditions outlined in the rule. In this case, the user is told that the message would be forwarded to Bob Jones. The system does not go on to describe other actions that might be performed on other types of messages from John Doe. It may be

the case that the user has little knowledge of the rule base but is just interested in the particular question. In this case, because the question is so specific and matches directly with a rule, it is preferable not to provide additional information which the user would probably not find relevant.

If the user poses a query: "What if I am on vacation and I get a message from John Doe?" the system provides information on rules that are closely matched. In this case the response would state that messages about information filtering would be forwarded to Bob Jones. Additional information would also be included about actions to be performed on other types of messages from John Doe.

If the user poses the query "what if I get a message from Dept. 9W21?" there may be a large number of rules that are partially matched by this condition, some close matches, others not close. In this case, details are provided on the most closely matched rules, and reference is given to the relevant conditions in rules that are not closely matched.

Using an assumption of user intent based on the nature of the query, ISCREEN seeks to avoid withholding important information, while not providing irrelevant information to the user.

Quality: The explanation should not present information that is false or for which there is inadequate evidence.

It is possible for a system to give wrong answers if it does not have access to necessary information, or it does not use enough inference in producing its responses. The following example illustrates: The user provides the single instruction that if a message is received from Toronto, then it should be filed in the "Toronto" box. If the user asks the system what would be done if a message is received from department 9w21, the system could answer that it would not do anything because it has not received any instructions regarding messages from this department. However if it is true that all members of department 9w21 work in Toronto, then each message received from the department would be filed in the "Toronto" box, contrary to what was stated in the answer to the query.

The screening would be performed properly because the system can examine an envelope to determine the department and location that it is from, but if it lacks the knowledge of relationships between the attributes that it uses to screen messages (departments, locations, job positions, etc.) then it will not be able to correctly answer hypothetical questions.

ISCREEN has the ability to recognize matches between conditions such as those noted in the above example and report on actions that would be taken. Inclusion of this ability is based on an assumption regarding user intent. It is assumed that the user's intention in posing the query is to find out what the system would actually do, as opposed to what the user told the system to do.

This approach to answering questions was adopted for two reasons. First, it is simple for users to see what they have instructed the system to do. To do this it is only necessary to look at the rules. On the other hand it is more difficult to predict the actions of the system when matching of variables is taken into account. The explanation system was designed to provide the information that is more difficult for users to get on their own. The other reason for this strategy

was to prevent ISCREEN from providing false information to users, as would have occurred in the above example.

Another requirement related to the Quality maxim is the ability to block false implications. The following two examples illustrate situations in which false implications should be blocked by the system:

1. The user instructs the message screener that any message received from John Doe is considered to be Important and should be filed in the "important" box. If the user then asks the message screener what it would do if the user were to receive a message from John Doe on the subject of trucks, the system could correctly respond that this would be an important message, and it would be filed in the "important" box. This may lead to the false implication that trucks are important. A response that would block this implication would be "I would file this message in your "important" box. Whether the message is on the subject of trucks is irrelevant."
2. The user may instruct the message screener that any messages received from directors should be filed in the "directors" box, and any messages received from Toronto should be filed in the "toronto" box. If the user then asks what would be done with a message received from a director in Toronto, the system could reply that it would be filed in the "directors" and "toronto" boxes. The query implies that the user believes that there are directors working in Toronto. The response implies to the user that this belief is correct. The response is correct in any case because if such a message were received it would be handled in the manner described. However, if it were true that there are no directors working in Toronto, the system should inform the user in order to block the false implication.

Relation: The explanation should be relevant to the context in which information is exchanged.

As outlined above, ISCREEN ranks the relevance associated with different items of information that could be provided on the basis of the relationship between the user's query and the rule base he is querying (an assumed level of user knowledge). Because the system responds to fairly specific questions in well understood contexts, it is relatively easy to ensure the relevance of the response.

Still, there are areas with a potential for misunderstanding or lack of clarity. For example, in screening messages ISCREEN often uses organizational information that the user may not be aware of. ISCREEN examines the relationship between the content in the response to a question and the content of the question to determine whether it would be relevant to expand on reasons for performing an action. The following example illustrates this point: A user specifies a rule called "file messages from managers" that states that if a message is received from a line manager it is to be filed in a "managers" box. If the user then asks what the system would do if the user were to receive a message from Bob Jones, the system could respond that it would be filed in the "managers" box because of the "file messages from managers" rule.

Because there is a discrepancy between what was specified in the rule (from a line manager) and what was specified in the query (from bob jones), it would

QUERY: Why was this message filed in the important box?

I filed this message in your important box because of the rule: 'handle important'. The rule states that if you were to receive a message that is 'very important' I should file that message in your important box. The message classification 'very important' relates to a message that is from one of your managers, rick wilson, janie newton, or bob johnson. This rule was executed because the message is from dave roberts (because dave roberts is one of your managers).
I also printed the message because of the rule 'auto print'.

Fig. 5. Explanation for "Why?" query.

probably also be relevant to add that the action would be taken because the message is from Bob Jones who is a line manager.

The ability to recognize potential matches between conditions specified in user queries causes ISCREEN to violate the relation maxim under certain conditions. In the above example it was worthwhile saying that a message from Dept 9w21 would be filed in the "toronto" box because any message from that department would also be from Toronto. If it were the case that the department had 200 members, with only 1 in Toronto, ISCREEN would also report that it may file the message in the Toronto box if it were from that member. Currently, ISCREEN does not have the ability to distinguish between matches that are likely to occur and matches that are unlikely. Thus, some of the explanations refer to actions that would be taken under highly unlikely circumstances and thus have little relevance.

Manner: *The explanation should be clear, unambiguous, brief, and orderly.*

Text in explanations is ordered according to relevance such that the user can read the most relevant portions of the explanation first. The explanations are provided in English and make use of pronominalization and coordination to avoid stylistic redundancy.

Examples of ISCREEN responses to *why* questions are illustrated in Figure 5. Examples of responses to *what if* questions are illustrated in Figure 6.

4. ISCREEN IMPLEMENTATION

4.1 Programs

The ISCREEN prototype is implemented on a VAX 750, using Franz LISP and Waterloo UNIX PROLOG.[1] LISP was used to implement the forms and screen management utilities used in the interface, while the core functionality of the system was implemented using PROLOG. To integrate ISCREEN with the existing Office Information System, two new types of programs were added: a Screening program and an Interface program.

There is one instance of the Screening program, which runs in the background and communicates with the OIS. This program intercepts messages as they are received by users of the OIS, compares the attributes of the message to the rules specified by the message recipient, and makes decisions about actions which

[1] UNIX is a trademark of AT&T Bell Laboratories.

QUERY: What if from(fred alkins), content(iscreen), copy(sam blake)

RESPONSE:

You have classified a message with these attributes as being 'toronto' and 'about ai'. I would file this message in your AI box. In this case it is not significant whether the message is copied to sam blake.

The rule: 'handle ai' would be responsible for this action. This rule states that if you were to receive a message that is 'about ai', I should file that message in your AI box. The message classification 'about ai' relates to a message that is from department 9w21 or stephen levitt, and with the content words iscreen, expertos, or tsf. This rule would be executed because fred alkins is in department 9w21.

The rule 'handle other toronto' would recommend filing the message in your other-tor box. This action would not be taken because you specified that the 'handle ai' rule should take precedence in this situation.

QUERY: What if from(robert sexton)

RESPONSE:

You have classified a message with this attribute as being 'outside toronto'. I would file this message in your impacts box, and print it.

The rule: 'handle impacts' would be responsible for this action. This rule states that if you were to receive a message that is 'outside toronto', I should file that message in your impacts box and print it. The message classification: 'outside toronto' relates to a message that is not from location tor.

Depending on other conditions, other actions may also be performed. If you were on vacation, I would forward this message to robin frank according to the rule 'forward impacts when on vacation'.

Fig. 6. Explanations for "What if?" queries.

should be taken on the message. The actions are then taken by the OIS. The decision to implement a single centralized screening program for the entire system, as opposed to dedicating a separate screening program to each user of the system, was based on practical issues of integrating the system with the OIS. A single task on the OIS picks up messages that are sent by the OIS users. This task was modified to pass the messages to ISCREEN and to complete delivery based on instructions provided by ISCREEN.

The interface program is invoked when users select the ISCREEN option from the messaging system. A single instance of the interface program is dedicated to each active user and handles modification of and queries against the user's rule base. When users modify their rules, the background program is notified so that it can load a new copy of the rule base.

The components of the ISCREEN system are illustrated in Figure 7. They are split into Processing components and the Knowledge Bases which they utilize.

4.2 Knowledge Bases

The User Knowledge Base contains information regarding each user of ISCREEN and the rule and message classification definitions entered by each user.

The Activity History is a record of actions that ISCREEN has taken on behalf of each user and the reasons for performing each action. ISCREEN maintains a

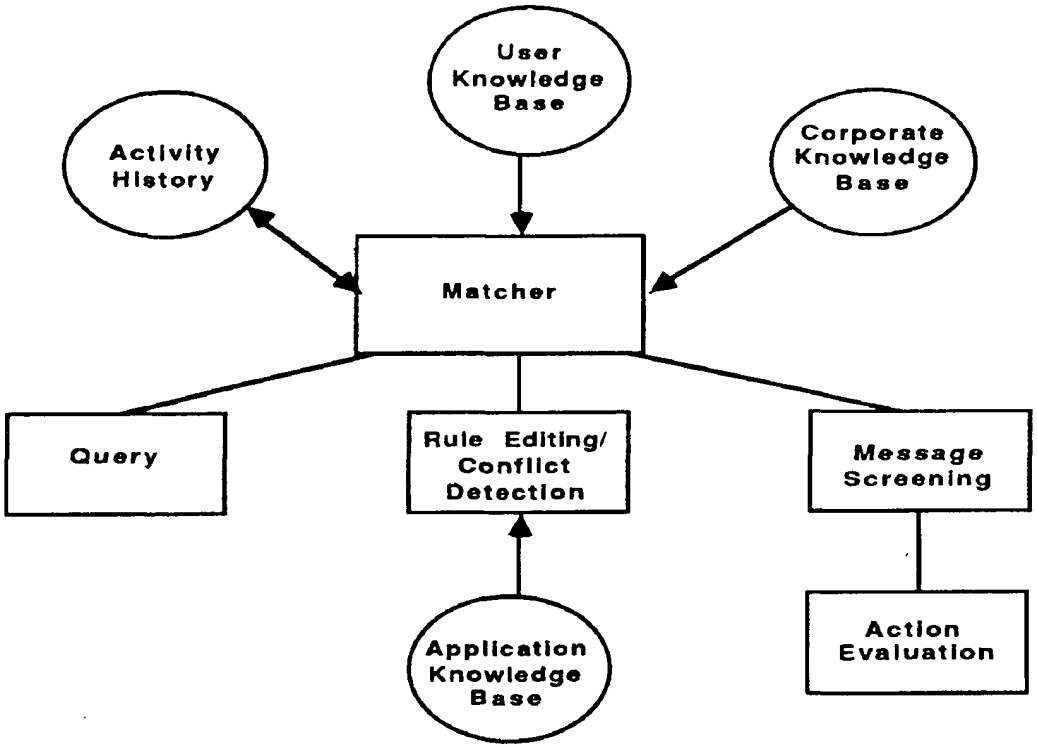

Fig. 7. ISCREEN COMPONENTS.

record of the reasons for performing each action to support the ability to answer questions about why the action was performed. Because the information that the decision was based on may change between the time that an action is taken and the time that a user poses a query, it is necessary to keep track of the reasons. Currently ISCREEN does not track changes in rules for the purposes of maintaining the correctness of explanations. If a rule has been changed since performing an action, ISCREEN generates an explanation providing the correct reasons, but references a rule which may not now recommend the action.

The Corporate Knowledge base provides information about the structure of the organization. The information includes the names of all users of the messaging system, their departments, locations, the companies they work for, the employees who report to them, and the employees that they report to. It is the Corporate Knowledge base that makes the use of *organizational information* variables possible. The data in the Corporate Knowledge Base is extracted from two existing corporate personnel databases and merged into a file. The file stores a record for each employee and provides a set of pointers indicating fellow group members, immediate managers, and subordinates. The knowledge base is kept up to date by extracting data on a regular basis from the personnel databases and using utilities to remerge the data. ISCREEN uses a set of PROLOG predicates which call subroutines written in C to execute queries.

The Application Knowledge base contains the rules that apply the sensibility check to actions and prevents conflicting actions from being taken. These rules

are applied in order to veto decisions that have been made by ISCREEN based on the user's rules.

4.3 The Matcher

The central processing component of ISCREEN is the Matcher. This component is used in generating explanations, in performing conflict detection, and in message screening. The purpose of the Matcher is to determine how a new rule entered by the user (in conflict detection), or a new message received by the user (in message screening), or a question asked by the user (in generating explanations), is related to the rule base defined by the user. Messages, queries, and rules are represented in the same format by ISCREEN. The Matcher accepts a single *entity* (either message, rule, or query) and compares it to each of the rules defined by the user, creating a "Match Profile" for each rule describing how the rule compares to the *entity*. The processes of conflict detection, message screening, and explanation are then completed using information left behind in the Match Profiles.

The Match Profile is composed of information about attributes that match, attributes that mismatch, and attributes that are irrelevant in both the *entity* to be matched, and the rule that it is compared against. Table I illustrates the contents of the match profile that is created when comparing the *entity*:

from(john doe, my manager), subject(AI), length(> 23)

to the rule conditions:

from(bob jones), subject(schedules), recipients(< 3)

Assuming that Bob Jones is the user's manager, the conditions from(bob jones) and from(my manager) are identified as matching. The "subject" conditions specified in the entity and the rules mismatch (AI vs schedules) and so are identified as mismatched conditions in both the entity and the rule. The "length" attribute specified as a condition in the entity was not referenced at all among the rule conditions, and so is identified as irrelevant. The same is true for the "recipient" attribute listed as a condition in the rule.

Depending on the context in which the Matcher is running, different criteria are used for judging whether two conditions match and for deciding when to give up trying to match an entity to a rule. When screening a message, the matcher is only interested in finding rules in which there are no mismatched or irrelevant conditions. In contrast, when the matcher is called for an explanation, an attempt is made to also find rules that "almost" match. The Matching criteria used in conflict detection are broader than those used in explanation and message screening because conflict detection attempts to deal with potential situations. For example: If the rule condition "content(schedules)" is matched against a message in message screening, the condition will fail if the word "schedules" is not found in the content of the message being screened. However, if this rule is compared to the rule condition "content(projects)" in conflict detection, it is not necessarily a mismatch because both the words "schedules" and "projects" could potentially appear together in a message.

Table I.

Matched Condition	from(bob jones, my manager)
Mismatched Entity Condition	subject(ai)
Irrelevant Entity Condition	length(> 23)
Mismatched Rule Condition	subject(schedules)
Irrelevant Rule Condition	recipients(< 3)

4.4 Message Screening

The message screening component operates in four stages: Matching, Action Recommendation, Conflict Resolution, and Sensibility. In the Matching stage information about a newly received message is compiled into a consistent format, and the Matcher is invoked to compare the message to the user's rule base. In the next stage the resulting Match Profiles are examined to determine which of the user's rules should fire. The screening component then recommends the actions listed in the fired rules. In the Conflict Resolution stage the system examines all of the recommended actions to determine if there is any conflict. The criteria for a conflict are defined in the Application Rule Base. The following are examples of actions which would be taken in Conflict Resolution:

—All duplicate actions are removed. Thus if two rules recommend forwarding a message to John Doe, only one of the recommendations is kept.
—If two rules match on the same message, and one was assigned precedence by the conflict detection component, actions recommended by the rule with lower precedence are ignored.
—If one of a user's rules recommends deleting a message and another recommends filing it, then the delete action is ignored because it cannot be undone.

When the system has finished with Conflict Resolution, the Sensibility check is performed. This involves using a set of rules from the Application Knowledge Base to eliminate or modify actions that do not make sense. Examples of rules are

—If a rule recommends forwarding a message to a user who is on the sender or recipient list, then do not forward it.
—If an action recommends replying to a message with over ten recipients, then use the "reply only" command which sends the reply only to the sender and not to all individuals copied on the message.

An interface should be available to allow definition and modification of such rules either on a per-user or corporate basis. This is not implemented in the current version of ISCREEN.

When the message screening component has completed its operation, it leaves three types of information which are maintained from one session to the next:

—Descriptions of actions that should be performed, and the names of the rules that recommend them. These are utilized by the Action-Taking component and also by the explanation component to list which actions were taken, and why they were taken.

—The Match Profiles of rules that fired. These are used by the Explanation System so that questions regarding why actions were taken can be answered.
—Information regarding actions that were vetoed because of Conflict Resolution and Sensibility checks. This information is used by the Explanation System in answering *why* questions.

4.5 Explanation

The explanation component consists of three different stages: Fact Generation, Conceptual Formulation, and Linguistic Realization. A user model is utilized in making decisions regarding information to be included in the response. This is similar to the approach used in the Text [11] and Ana [7] text generation systems.

ISCREEN's user model is generated on the basis of the correspondence between the user's query and the rule base. This correspondence determines the amount of information that will be included in the response. An additional part of the user model is the assumption that users expect ISCREEN to respond to *what if* with information about all events which could occur, rather than echo rules which directly match the query. These assumptions are explained in Section 3.4.

The Fact Generation stage collects and formats information under a number of set categories that could be provided to the user in an explanation. Different types of information are collected depending on the type of question asked. To answer *why* questions the Fact Generator makes use of information left behind by the Message Screening component that is maintained in an activity history from one session to the next. This information includes a description of the action performed on a message and the rule which recommended it; other actions that were performed on the same message; the match profile describing the correspondence between the message and the rule which caused ISCREEN to act on it; and rules which also matched on the message but were not followed because other rules took precedence.

To answer a *what if* question, the Message Screening component is invoked with a hypothetical message that has the attributes described in the user's query. For example, if the user asks "what if I get a message from John Doe on the subject of project schedules?" the message screener is invoked with a hypothetical message from John Doe about project schedules. This causes the Matcher to be invoked followed by Action Recommendation, Conflict Resolution, and Sensibility. The Matcher is run in an explanation mode so that the results are only kept while the explanation is being produced, and the matching and stopping criteria are different from those used in actual message screening. When the hypothetical message has been screened, information is left behind in the system's workspace describing actions that would be performed, the rules which recommend the actions, and the reasons that the rules fired. Match Profiles of rules that would "almost" fire are also stored, along with information regarding rules that were ignored because of Conflict Resolution and Sensibility.

The Conceptual Formulation stage selects and orders a set of facts generated in the previous stage and constructs the format of the explanation. At the top level, there are a set of schemata which describe how the discourse should be ordered. There is one schema for each type of explanation produced (why, what

if, conflict detection, rule description). The schemata were designed by writing a number of sample explanations and analyzing the structure of those which were judged to be most understandable. The following is the structure of the schema used for producing responses to *what if* questions:

1. Describe message classifications that are matched by the query
2. Describe actions that would be performed on the message
3. Describe Conditions specified in the query which are irrelevant to the actions taken
4. Describe the rules that would recommend the actions
5. Describe conflicts that would occur between rules
6. Describe rules that would recommend other actions given the truth of conditions not referenced in the query.

The schemata call templates which are used to organize text for each of the categories described above. There may be a number of templates defined for each category, each one producing different results. The templates are selected on the basis of facts produced in the Fact Generation stage. For example, in an explanation for which there is a lot of information to provide for the second category, a schema may be selected that produces no text for the sixth category (i.e., if there is a lot of highly relevant information, do not display information of less relevance). In some cases, templates are selected on the basis of random variables to provide variety in the wording of responses. The templates make calls to components at the Linguistic Realization stage to fill in the actual words to be used in the response.

The Linguistic Realization stage deals with the construction of sentences and is largely independent of the screening application. This stage handles conjugation of verbs, declination of nouns, insertion of punctuation and conjunctions, pronominalization, and coordination.

4.6 Conflict Detection

The Conflict Detection component is invoked when the user saves a new rule definition. This component runs the message screening component on a hypothetical message with the attributes described in the rule definition. The resulting match profiles describing the correspondence between the new rule and each of the existing rules are then examined to find potential conflicts. The information in the match profiles is used by the explanation component to describe the nature of each conflict. For each conflict found, ISCREEN either assigns precedence (when the match profile indicates that one rule describes messages which are a subset of those described by the other rule) or asks the user which of the rules should take precedence.

5. CONCLUSIONS

The development of information filters which would act as intelligent assistants to office workers could potentially help users to cope with the large volumes of information handled within office information systems. The ISCREEN message screening system was found to effectively screen text messages received by a small group of users of an office information system. It was found that, in order

to screen messages effectively, ISCREEN requires knowledge about the organization in which it operates and requires the ability to use that knowledge flexibly in completing a number of different tasks.

The interface of an information filtering system must be simple enough to allow end users to change rules on a regular basis. This includes not only ease of use but the ability to accept incomplete or conflicting instructions and request clarification from the user. Because filtering systems act independently of the user, they require the ability to communicate effectively in describing situations which have occurred, in describing situations which may potentially occur, and in asking for clarification of conflicting instructions. An explanation component, which is flexible and cooperative, is needed. In the ISCREEN system, text generation techniques were found to be very useful in implementing the explanation component.

The true utility of ISCREEN remains to be tested. The system is currently limited in its ability to understand the content of messages and would certainly benefit from the application of text understanding techniques. However, in producing a tool that could actually be integrated into an office system, the value of text understanding would have to be weighed against the performance requirements for the system. Testing the utility of this tool will require a trial with a larger user group accessing ISCREEN on a daily basis.

ACKNOWLEDGMENTS

I am indebted to Richard Goodwin and Jay Gauthier of Bell-Northern Research for their ideas and assistance in implementing the ISCREEN system.

REFERENCES

1. GRICE, H. Logic and conversation. In *Syntax and Semantics: Speech Acts.* Vol 3, P. Cole and J. Morgan, Eds., Academic Press, Orlando, Fla., 1975, pp. 41–58.
2. JOSHI, A., WEBBER, B., AND WEISCHEDEL, R. Living up to expectations: Computing expert responses. In *Proceedings of the American Association for Artificial Intelligence-8* (Austin, Tx., Aug. 1984).
3. KUKICH, K. Knowledge-based report generation. Ph.D. dissertation, Carnegie-Mellon University, Pittsburgh, Pa., May 1984.
4. MALONE, T., GRANT, K., TURBAK, F., BROBST, S., AND COHEN, M. Intelligent information sharing systems. MIT Industrial Liaison Program Report, 3-36-87, Massachusetts Institute of Technology, Cambridge, Mass.
5. MCKEOWN, R. *Text Generation.* Cambridge University Press, Cambridge, England, 1985.
6. MYER, T. H. Future message system design: Lessons from the Hermes experience. In *Proceedings of Distributed Computing COMPCON 80* (Washington, D.C., Sept.). IEEE, New York, 1980, pp. 76–84.

BIBLIOGRAPHY

BARBER, G. Supporting organizational problem solving with a work station. *ACM Trans. Off. Inf. Syst. 1*, 1 (1983), 45–67.

COHEN, P., DAVIS, A., AND DAY, D. Representativeness and uncertainty in classification system. *The AI Magazine*, (Fall 1985), 136–149.

CROFT, B., AND LEFKOWITZ, L. Task support in an office system. *ACM Trans. Off. Inf. Syst. 2*, 3 (July 1984), 197–212.

FIKES, R., AND HENDERSON, D. On supporting the use of procedures in office work. In *Proceedings of the First Annual American Association for Artificial Intelligence Conference*, 1980, pp. 202–207.

MALONE, T., GRANT, K., AND TURBAK, F. The Information Lens: An intelligent system for information sharing in organizations. In *Human Factors in Computing Systems: CHI '86 Conference Proceedings* (Boston, Apr. 13–17). ACM, New York, 1986, pp. 1–8.

MANN, W. Discourse structures for text generation. USC/Information Sciences Institute, Tech. Rep. RR-84-127, Marina del Rey, Calif., Feb. 1984.

MERCER, R., AND ROSENBERG, R. Generating corrective answers by computing presuppositions of answers, not of questions. In *Proceedings of Conference of Canadian Society for Computational Studies of Intelligence* (University of Western Ontario, London, Ontario, May). Canadian Information Processing Society, Toronto, 1984, pp. 16–19.

ZLOOF, M. Office-by-example: A business language that unifies data and word processing and electronic mail. *IBM Syst. J. 21*, 3, (1982), 272–304.

Received February 1986; revised October 1987; accepted February 1988